GOA BOARD OF SECONDARY AND HIGHER SECONDARY EDUCATION
Alto Betim, Bardez, Goa 403 521
(A Corporate Statutory Body constituted by an Act of the State Legislature)
www.gbshse.gov.in sec-gbshse.goa@nic.in 8459710179

GBSHSE/ACAD/ICT/2024-25/ Date: 05/07/2024

Circular No: 44

To,
The Heads of all Recognised Secondary
Schools Under the jurisdiction of this Board

Sub: Introduction of HTML and CSS in the ICT Syllabus for Class X for the Academic year 2024-25 regarding…

Sir/Madam,

With reference the above, it is to inform you that Board of Studies in ICT has decided to introduce HTML5 and CSS3 as part of the Information and Communication (ICT) syllabus for Std. X, for the academic year 2024-25. In view of implementation of NEP20 for Class IX and in the light of Circular 33 dated 28/06/2022.

This initiative aims to equip our students with essential web development skills, ensuring they are well-prepared for the digital age.

The contents of the New ICT Syllabus are as follows:

1. **Introduction to HTML**

- Basic structure of an HTML document
- Common HTML tags and their usage
- Creating hyperlinks, lists, and tables.

2. **Introduction to CSS**

- Understanding the role of CSS in web design
- CSS syntax and selectors
- Styling text, layouts, and backgrounds

3. **Practical Sessions**

- Developing simple web pages using HTML and CSS
- Projects and assignments to reinforce learning

The detailed syllabus is available on the official website at www.gbshse.gov.in.

Therefore, All Heads are hereby enjoined upon to inform the concerned and take appropriate action.

--2--

Further, a copy of this circular be displayed at a prominent place of school notice board.

If schools have any queries and further clarification, may send the same through email ID, sec-gbshse.goa@nic.in.

05/7/24

(Vidhyadatta B. Naik)
Secretary

Encl: Detailed ICT Syllabus for Std. X (Available on the Official Website)

Copy to:

1. The Director, Directorate of Education, Porvorim-Goa
2. The Director, SCERT, Porvorim-Goa
3. All the Section Heads of this Office.
4. Shri Dinesh S. Zalmi, Convener, BOS(ICT), PES Shri Ravi Sitaram Naik Higher Secondary School Higher secondary school, Farmagudi-Goa

12

HTML 5

A coursebook of ICT

Goa Board of Secondary & Higher Secondary

Alto - Betim, Bardez, Goa 403 521

WHAT YOU NEED TO KNOW

- Introduction
  - Internet
    - Presentation of Information in WWW
- Websites
  - Website
  - Web Pages
- What is HTML?

HTML BASICS

- Understanding HTML Environment?
  - Text Editor
  - Browsers

- Basic Structure of HTML
  - Creating Basic Structure of HTML
    - *Head section*
    - *Body section*
  - Tags and elements
    - *Rules for writing tags*
    - *Container tags*
    - *Empty tags*
  - Saving File
  - Open file in browser
  - Open file for editing

- Attributes and values
  - Describing attributes
  - Length unit identifiers
  - Color values

- Using Attribute
  - Example: Background Color

Types of HTML Elements
- Basic HTML elements
- Block-level elements
- Inline elements

HTML ELEMENTS

Basic HTML elements
- Head
- Title
- Body
  - *Text color*
  - *Background color*
  - *Background image*
  - *Margins – top margin, left margin*

Block-level elements and their attributes
- Header Levels 1 through 6
  - *Align*
- Paragraph
  - *Align*
- Marquee
  - *Direction*
- Horizontal Rule
  - *Size*
  - *Width*
  - *Color*
  - *Noshade*
  - *Align*
- Lists
  - *Ordered Lists*
    - *Start attribute*
  - *Unordered Lists*
    - *Value attribute*
- Table

Table structure
Table border
Table border color, background color,
Cellpadding, Cellspacing
Colspan, Rowspan
Caption
Th border, Th border color, Th background color,
Tr border, Tr border color, Tr background color,
Td border, Td border color, Td background color,
Table width, align

Text Decoration and styling

Bold – b
Italic – i
Underline – u
Deleted text – del
Big – big
Small – small
Strong – strong
Subscript – sub
Superscript – sup
Quotation mark – q
Preformatted text – pre
Font
Color
Size
Face
Line break – br

Inline elements and their attributes

Image
Src, alt, align, vspace, hspace
Audios
Videos

Links

PROJECT WORK

Teachers Guidance

Data collection

Layout

12

CSS 3

A coursebook of ICT

Goa Board of Secondary & Higher Secondary

Alto - Betim, Bardez, Goa 403 521

AUTHOR

Dinesh S. Zalmi

REVIEW

Ajay Jadhav

Suraj S. Naik Gaonkar

Goa Board of Secondary & Higher Secondary

Alto - Betim, Bardez, Goa 403 521

CONTENTS AT GLANCE

3. Grouping Selectors
4. Multiple Style Rules

IV. Ways to Use CSS

1. Inline Style
2. Internal Style sheet
3. External Style sheet
4. Using Multiple Style sheets

V. CSS Properties

1. Color
2. Background
3. Text
4. Font
5. Lists
6. Table
7. Link

VI. Box Model

1. Height and Width
2. Padding
3. Border
4. Margin
5. Outline

VII. Layout Properties

1. Display/Visibility
2. Position
3. Float/Clear

PART - I

WHAT YOU NEED TO KNOW

1. Understanding CSS
2. What is CSS?
3. Advantages of using CSS
4. CSS Versions

UNDERSTANDING CSS

HTML was never intended to contain tags for formatting a web page. It was created to describe the content of a web page, like: h1, p, etc.

Tags like `<font>`, and color attributes were added to the HTML 3.2 specification.

Development of large websites, where fonts and color information were added to every single page, became a long and expensive process. To solve this problem, the WWW Consortium (W3C) created CSS.

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

WHAT IS CSS?

CSS stands for "**Cascading Style Sheets**".

"**Cascading**" refers to the procedure that determines which style will apply to a certain section, if you have more than one style rule. The "**style**" is referred to the look of a certain part of your page. The "**sheets**" are like templates, or a set of rules, for determining how the webpage will look.

CSS is intended to simplify the process of making pages presentable. It handles look and feel part of webpage. You can use CSS to control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used,

as well as a variety of other effects. It is also used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

So, CSS is a styling language that contains a set of rules to tell browsers how your webpage should look.

CSS provides a powerful control over the presentation of an HTML document on screen, on paper, or in other media.

ADVANTAGES OF CSS

1. CSS saves time:

   Styles defined for the HTML elements in a CSS file can be applied to as many web pages as you want.

2. Pages load faster

   Just write one CSS rule of a tag and apply it to all the occurrences of that tag in HTML document. So, less code; means faster download times.

3. Easy maintenance

   To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.

4. Superior styles to HTML

   CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

5. Multiple Device Compatibility

Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for hand-held devices such as PDAs and mobile-phones or for printing.

6. Global web standards

Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

CSS VERSIONS

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags. Today CSS has various levels, each level of CSS builds upon the last, typically adding new features and typically denoted as CSS 1, CSS 2, CSS 2.1, CSS 3, etc.

CSS also have subset of one or more levels of CSS built for a particular device or user interface, called Profiles.

PART - II

CSS BASICS

1. Syntax
2. Selector
3. Declaration
4. Property values
5. Length Unit identifiers
6. CSS Comments

SYNTAX

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of two parts, (1) Selector, and (2) Declaration

1. Selector

   A selector is an HTML element you want to add style to. This could be any like **<h1> <p>** or **<table>** etc.

2. Declaration

   The declaration is the statement of style for that element. It is made up of property and value.

   Property: A property is the aspect of an element you want to change; for eg. **color**, **border**, **font**, etc. Put simply, all the HTML attributes are converted into CSS properties.

   Value: Value is the exact setting for the property. For example, **color** property can be set to value **red** or the **font-size** can be set to **30px**, and so on.

   CSS Style Rule Syntax is as follows:

   ```
   Selector {declaration;}
   Or
   Selector {property1:value;}
   ```

   Example:

   The style rule for **<h1>** element can be defined as follows:

   ```
   h1{text-align:center;}
   ```

In this example, **<h1>** is a selector, **text-align** is a property and **center** is the value of the property.

Another Example:

```
body{background-color:green; Margin-left:2cm;}
```

When you add **more** properties to the style rule, the same are separated by semi-colon. (**;**). In the above example, **background-color, and Margin-left** are separated by semicolon.

PROPERTY VALUES

Property values are of three types, viz. specified, computed and actual

1. Specified

   It is the value specified in declaration. This type of value is sub-divided as absolute, and relative.

   Absolute*:* value can be determined without reference to context (e.g., 2cm)

   Relative*:* value depends on context (e.g., larger)

2. Computed

   It is absolute representation of relative value (e.g., larger might be 1.2 x parent font size)

3. Actual

   It is the value actually used by browser (e.g., computed value might be rounded)

CSS LENGTH UNIT IDENTIFIES

Identifier	Meaning
in	- inches
cm	- centimeters
mm	- millimeters
pt	- points: 1/72-inch
pc	- picas: 12 points
px	- pixels: typically 1/96-inch
em	- 1em is roughly the height of a capital letter in the reference font
ex	- 1ex is roughly the height of the lowercase 'x' character in the reference font.

CSS COMMENTS

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines.

Example of single line comment:

```
p {
/*The paragraph elements will appear in red color.*/
color: red; }
```

Example of multi-line comment:

```
p {
/*The paragraph elements in this web page
will appear in red color.*/

color: red; }
```

PART - III

SELECTORS

1. Selector Types
2. Selector Combinators
3. Grouping Selectors
4. Multiple Style Rules

SELECTOR TYPES

There are three types of Selectors

1. Element Selector
2. ID Selector, and
3. Class Selector

1. The Element Selector

The element selector selects the elements based on the element name. For example, with a single declaration, all **<h1>** elements on a page can be selected and the style rule is defined to them as below.

```
h1{color: red; text-align:center;}
```

Level 1 header, i.e. **<h1>** is described with **color** attribute set to **red** and **text-align** to the **center**.

2. The ID Selector

You can define style rules based on the **id** attribute of the elements. All the elements having that id will be formatted according to the defined rule.

```
#black {color: #000000;}
```

This rule renders the content in black for every element with **id** attribute set to **black** in our document. You can make it a bit more particular. For example:

```
h1#black {color:#000000;}
```

This rule renders the content in black for only **<h1>** elements with id attribute set to black.

The true power of **id** selectors is when they are used as the foundation for descendant selectors. For example:

```
#black h2 {color:#000000;}
```

In this example, all level 2 headings will be displayed in black color when those headings will lie within tags having **id** attribute set to black.

3. The Class Selectors

The style rules defined in the class selector can be applied to any element using class attribute for the element.

```
.black {color: #000000;}
```

This rule renders the content in black for every element with class attribute set to **black** for the element. You can make it a bit more particular. For example:

```
h1.black {color:#000000;}
```

This rule renders the content in black for only **<h1>** elements with class attribute set to black.

You can apply more than one class selectors to a given element. Consider the following example:

```
<p class="center bold"> This para will be styled by
the classes center and bold. </p>
```

CSS COMBINATORS

Combinator is something that explains the relationship between the selectors. A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

CSS combinators are of following types:

1. Descendant selector,
2. Child selector,
3. Adjacent sibling selector,
4. General sibling selector,
5. Universal selector, and
6. Attribute selector.

1. Descendant Selector

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element, you use descendant selector. It matches all elements that are descendants of a specified element. As given in the following example, the style rule will apply to `li` element – only when it lies inside the **<ul>** tag.

```
ul li {color: #000000;}

div p {background-color: yellow; }
```

2. Child Selector

The child selector selects all elements that are the immediate children of a specified element. The following example selects all **<p>** elements that are immediate children of **<body>** element. Other paragraphs put inside other elements like ** ** or **<td>** would not have any effect of this rule.

```
body > p {color:#000000;}
```

3. Adjacent Sibling Selector

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element. Sibling elements must have the same parent element, and “adjacent” means “immediately following”.

The following example selects all <p> elements that are placed immediately after elements:

```
Div +p {background-color:yellow;}
```

4. General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all <p> elements that are siblings of elements:

```
Div ~p {background-color: yellow;}
```

5. The Universal Selector

Rather than selecting elements of a specific type, the universal selector selects the name of any element type:

```
* {color:green;}
```

This rule renders the content of every element in our document in black color.

6. The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text:

```
hr[color="red"]{width:50%; size=40;}

h1[color="red"]{text-align: center; color: green;}
```

The advantage to this method is that, the same type of element with other value to the attribute is unaffected.

There are following rules applied to attribute selector:

`H1[color]` - Selects all Level 1 Header elements with a color attribute.

`H1[color="red"]` - Selects all Level 1 Header elements whose color attribute has a value of exactly "red".

`H1[color~="red"]` - Selects all Level 1 Header elements whose color attribute contains the word "red".

`h1[color|="red"]` - Selects all Level 1 Header elements whose color attribute contains values that are exactly "color", or begin with "color".

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example:

```
h1, h2, h3   {
              color:#36C;
              font-weight:normal;
              letter-spacing:.4em;
              margin-bottom:1em;
              text-transform:lowercase;
             }
```

This define style rule will be applicable to `h1, h2` and `h3` element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example:

```
h1 {  color:#36C;
      font-weight:normal;
      letter-spacing:.4em;
      margin-bottom:1em;
      text-transform:lowercase;
   }
```

Here all the property and value pairs are separated by a semicolon (;). You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

PART - IV

WAYS TO USE CSS

1. Inline Style
2. Internal Style sheet
3. External Style sheet
4. Using Multiple Style sheets

Where do I put my style rules?

CSS style rules can be written at three different places. According to their functionality, these style-rule-places:

1. Inline style
2. Internal Style Sheet
3. External Style Sheet

1. Inline style

Inline style is used to apply unique styles for a single element. It uses **`style`** attribute to declare the style. The value of **`style`** attribute is a declaration. See the below given code.

```
<h1 style="color: SteelBlue;"> Inline styling. </h1>
```

Declaration describes the styles to be applied to the **`<h1>`** element.

Sample program

```
<html>
<head>
 <title> Inline styling</title>
</head>
<body>
 <h1 style="color: SteelBlue;"> Inline styling. </h1>
</body>
</html>
```

2. Internal style sheet

Internal style sheet is used to apply unique style in the current page. It uses the **`<style>`** element in **`<head>`** section of HTML document to describe style rules. See the below given code.

```
<head>
     <style>
          h1 {color:red; font-family:verdana;}
     </style>
</head>
```

Sample program

```
<html>
 <head>
     <title> Internal styling</title>
     <style>
          h1 {color:red; font-family:verdana;}
          p {color:darkorchid; font-size:16;}
     </style>
 </head>
 <body>
     <h1>Red color heading .</h1>
     <p>Font size and color of the text displayed in
     default values.</p>
 </body>
</html>
```

3. External style sheet

External style sheet method calls the style rules from a separately stored css file using the **<link>** element in **<head>** section of HTML document. CSS file is a separately stored file with extension name **.css** that holds style rules for different types of HTML elements. See the below given code.

Sample code

Code in CSS file **'style1.css'**

```
h1, p {
      text-align:center;
      color:red;
      }
```

Code in HTML file **'MyPage.html'**

```
<html>
<head>
 <link rel="stylesheet" type="text/css"
 href="style1.css">
</head>
<body>
 <h1>This is a heading</h1>
 <p>This is a paragraph</p>
</body>
</html>
```

Same CSS file can be referred in many pages. So, just one CSS file can change the look of an entire website.

Using multiple style sheets

If a selector (element) has different style rules defined using different style sheets then the value from the last read style sheet will be rendered by the browser.

For instance, when we write code in external style sheet, we provide link in head section of our HTML document as below:

```
<link rel=”stylesheet” type="text/css" href=”style1.css”>
```

- In the above example, **`href`** attribute declares the file **`“style1.css”`** which will be read first by HTML document and the system will apply it accordingly.
- Apart from the style referred in external style sheet, if you have declared specific style for the page using internal style method, the internal style file will be read next to external style, and if specific change is declared for some elements, that will be applied to those elements.
- The inline style, if you have used any, will be read after internal style.

So, if you are using multiple style sheets, program will read the code in the order – external style, internal style and inline style, and the value from the last read style sheet will be used.

PART - V

CSS PROPERTIES

1. Introduction
2. Color
3. Background
4. Text
5. Font
6. Lists
7. Table
8. Link

INTRODUCTION

CSS properties are the key to altering the styling of HTML elements in your web documents. They are specified in the CSS standard. They define how the styles should look on the Web page or elsewhere. Each property has a set of possible values. Some properties can affect any type of element, and others apply only to particular groups of elements.

The properties in this appendix are grouped into areas according to the type of element it affects; in fact, in many cases, one property affects another. The property groups include the following:

- Colours
- Background
- Text
- Font
- Lists
- Table
- Styling Links
- Box model
- Display
- Position
- Float

This part covers properties up to styling links. Other properties will be covered in next parts of this writing.

USING COLORS

Color

`Color` property is used to set the color of the text/element. Color is often specified by:

A Hex value – like "`#ff0000`"

An RGB value – like "`rgb(255,0,0)`"

A color name – like "`red`"

Sample code

```
<style type="text/css">
   Body {color:yellow;}
   H1{color:green;}
<style>
```

USING BACKGROUNDS

Background-color

Background-color property is used to change the background color of any element.

Value

Any color value like:

```
red
#663399
```

Sample code

```
<style =”text/css”>
     Body  {background-color:yellow;}
       H1  {background-color:blue;}
        P  {background-color:red;}
</style>
```

Background-image

Background-image property is used to set background image. By default, the image is repeated so it covers entire element. Value must be given as URL and the filename within the bracket.

Sample code

```
<style type=”text/css”>
     Body  {background-image:url(‘filename’);}
</style>
```

Image-repeat

Image-repeat property is used to repeat the image both horizontally and vertically.

Values:

Repeat-x : replicates image horizontally

Repeat-y : replicates image vertically

No-repeat : image won’t be replicated

Sample code

```
     Body  {Background-image:url(‘baby.jpg’);
           Background-repeat:repeat-y;}
```

Image-position

Image-position attribute is used to align the background image to left, right, top or center.

Values

right,

left,

top,

center

When you write value pair as **'0% 0%'** then the Image is placed in the upper left corner of the box

Possible combinations of keywords and their interpretations (e.g.).

'top left' and 'left top' same as '0% 0%'

'top', 'top center' and 'center top' as '50% 0%'

'right top' and 'top right' as '100% 0%'

'center' and 'center center' as '50% 50%'

'bottom left' and 'left bottom' as '0% 100%'

'bottom right' and 'right bottom' as '100% 100%'

Background-attachment

Background-attachment property is used to set image scroll or not to scroll with rest of the page.

Values

fixed,

scroll

Sample code

```
<style type="text/css">
    body { Background-image:url  ('baby.jpg');
    body { background-attachment : fixed ; }
<style>
```

Background

Background is a shortcut property used to set all background properties in one declaration. The list of order for specifying values is as follows:

```
background-color, background-image, background-repeat,
background-attachment, background-position
```

Sample code

```
<style type="text/css">
  body {background :"red" url("bg_image.jpg") no-repeat
  right top; }
</style>
```

TEXT

Letter-spacing

Letter-spacing property is used to set the space between the characters.

Sample code

```
H1 {letter-spacing:0.5cm;}
```

Word-spacing

Word-spacing property is used to set the space between the words.

Sample code

```
P {word-spacing:1cm;}
```

Line-height

The Line-height property is used to set the height of line.

Sample code

```
P {line-height:200%;}
```

Text-decoration

The Text-decoration property is used to underline text, put line over text or line through text. It can be set using pre-defined values: **Underline, Overline, Line-through, None** (to remove line)

Sample code

```
H3 {text-decoration:underline;}
```

Text-align

The text-align property is used to align the text to the left indent, right indent, center to the left indent and right indent or align to both left and right indent.

The alignment can be set by using one of the three pre-defined values: **right, left, center or justify.**

Sample code

```
H1{text-align:center;}
```

Text-indent

Text-indent property is used to indent the first line of text. It adds the specified space in the first line, before the text. It can be set by specifying the distance from the left margin.

Sample code

```
H1 {text-indent:1cm;}
```

Text-transform

The text-transform property is used to transform the text to uppercase,

lowercase, capitalize. It can be set using predefined values: **uppercase, lowercase, capitalize, none** (default)

Sample code

```
H1 {text-transform:uppercase;}
```

FONT

Font-family

The font-family tag sets the face of the font. Different values can be used; if browser does not support the first font, it tries the next font, and so on.

Sample code

```
H3 {font-family:verdana;}
p {font-family:cambria, verdana, tahoma;}
```

Font-size

The font-size property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs. The font-size value can be an absolute size, or relative size.

Absolute size sets the text to a specified size. Setting size with pixel gives full control over the text size. It does not allow a user to change the text size in all browsers.

Sample code

```
p {font-size:18px;}
```

Relative size allow users to resize the text (in the browser menu), many developers use em instead of pixels. The em size unit is recommended by the W3C.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px. The size can be calculated from pixels to em using this formula: pixels/16=em.

Sample code

```
p {font-size:1.5em;}
```

Font-style

The font-style property sets the style of the font. It is mostly used to italicize text. Font style can be set by using any of the predefined values: **Italic, Normal, oblique.**

Sample code

```
H3 {font-style:italic;}
```

Font-variant

The font-variant property is specifies whether or not a text should be displayed in a small-caps font. In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

Sample code

```
p.normal {font-variant:normal;}

p.small {font-variant:small-caps;}
```

Font-weight

The font-weight property specifies the weight of a font. Predefined values are normal, and bold. Also numeric value from 100 to 900 can be assigned to indicate the font weight.

Sample code

```
P {font-weight:500;}
```

Font

Font property sets all the font properties in one declaration. The order of values for font is:

```
font-style, font-variant, font-weight, font-size,
font-family
```

Sample code

```
p {font:italic bold 16px arial 400;}
```

LISTS

In HTML, there are two main types of lists:

1. Unorder lists (<ul>) – the list items are marked with bullets
2. Ordered lists (<ol>) – the list items are marked with numbers or letters

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

Different List Item Markers

List-style-type

The list-style-type property specifies the type of list item marker. The following example shows some of the available list item markers:

Sample code

```
ul.a {list-style-type:circle;}
ul.b {list-style-type:square;}
```

Values for unordered lists

```
circle, disc, square, none
```

Values for ordered lists

```
Decimal, decimal-leading-zero, upper-roman, lower-roman,
none
```

An Image As The List Item Marker

List-style-image

Sample code

```
ul {list-style-image:url("sqpurple.gif");}
```

Values

In the above example, '**sqpurple.gif**' is the name of image. Image size should be small.

Position The List Item Marker

List-style-position

Sample code

```
ul {list-style-position:inside;}
```

Values

```
Inside, outside
```

List – Shorthand Property

List

The list-style property is a shorthand property. It is used to set all the list properties in one declaration:

Sample code

```
ul {list-style: square inside url("sqpurple.gif");}
```

When using the shorthand property, the order of the property values are:

```
List-style-type, list-style-position, list-style-image
```

If any of the property values above are missing, the default value for the missing property will be inserted, if any.

Styling List With Colors

We can also style lists with colors, to make them look a little more interesting.

Anything added to the <ol> or <ul> tag affects the entire list, while properties added to the <li> tag will affect the individual list items. See the example below:

Sample code

```
ol { width:100%; background:#ff9999; padding:20px;}
ul { width:100%; background:#3399ff; padding:20px;}
ol li{background:#ffe5e5; padding:5px; margin-left:35px;}
ul li{background:#cce5ff; margin:5px;}
```

TABLE

Border

To specify table borders in CSS, use the border property.

The example below specifies a black border for <table>, <th>, and <td> elements;

Sample code

```
table, th, td {border:1px solid black;}
```

Border-collapse

The border-collapse property sets whether the table borders should be collapsed into a single border:

Sample code

```
table {border-collapse:separate;}

table, th, td {border:1px solid black;}
```

Values

```
collapse,

separate,

inherit
```

Table Width And Height

Width/Height

Width and height of a table are defined by the width and height properties.

The example helow sets the width of the table to 100%, and the height of the <th> element to 50px:

```
table { width:100%;}

th {height:50px;}
```

Horizontal Alignment

Text-align

The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.

By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned.

The following example aligns the text in <th> element to the left:

```
th{text-align:left;}
```

Vertical Alignment

Vertical-align

The vertical-align property sets the vertical alignment (like top, bottom and middle) of the content in <th> or <td>.

By default, the vertical alignment of the content in table is middle for both <th> and <td> elements.

The following example sels the vertical text alignment to bottom for <td> elements:

Sample code

```
td {height:50px; vertical-align:bottom;}
```

Border-spacing

Specifies the distance between the borders of adjacent cells.

Caption-side

Specifies the placement of a table caption relative to the table

Values

```
top (default),
right,
bottom,
left,
inherit
```

Sample code

```
Table {caption-side:bottom;}
```

Empty-cells

Specifies whether or not to display borders and background on empty cells in a table

Values

```
show(default),
hide,
inherit.
```

LINKS

Links can be styled differently depending on what state they are in.

The four states of links are:

`a:link`	–	a normal, unvisited link
`a:visited`	–	a link the user has visited
`a:hover`	–	a link when the user mouses over it
`a:active`	–	a link the moment it is clicked

Sample code

```
<style>
    a:link {color:red;}
    a:visited {color:green;}
    a:hover {color:hotpink;}
    a:active {color:pink;}
</style>
```

Remove Underline From The Link

The text-decoration property is mostly used to remove underline from links:

```
a:link {text-decoration:none;}
a:link {text-decoration:underline;}
```

Use Background Color For The Link

The background-color property can be used to specify a background color for links:

```
a:link {background-color:orange;}
a:visited {background-color:maroon;
```

PART - VI

BOX MODEL

1. Height and Width
2. Padding
3. Border
4. Margin
5. Outline

BOX MODEL

There are two types of elements in HTML:

1. Block-level elements, and
2. Inline elements.

1. **Block-level Elements -** A block-level element always starts on a new line and takes up the full width available. It stretches out to the left and right as far as it can by taking up the full width available.

   Example of block-level elements:

   ```
   <div>
   <h1> - <h6>
   <p>
   <form>
   <header>
   <footer>
   <section>
   ```

2. **Inline Elements -** An inline element does not start on a new line and only takes up as much width as necessary.

   Example of inline elements:

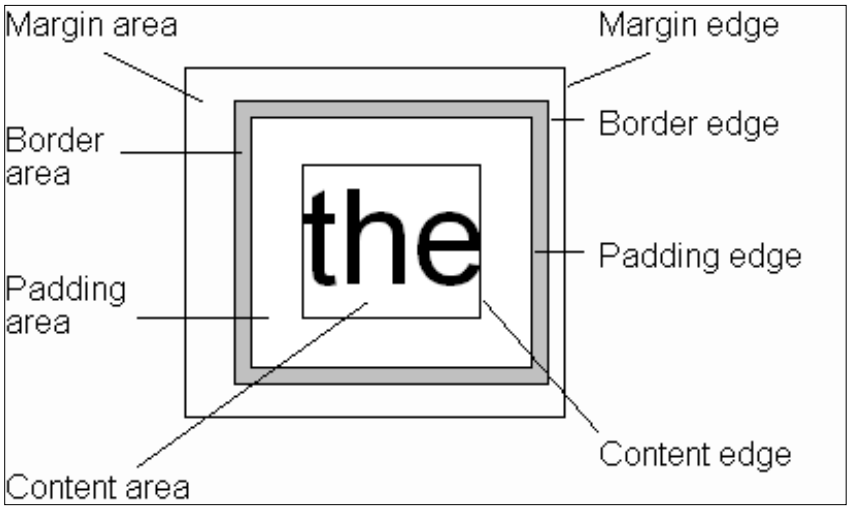
   ```
   <span>
   <a>
   <img>
   ```

CSS Boxes

Box generation is the part of the CSS visual formatting model that creates boxes from the document's elements.

The type of the box generated in CSS depends on the value of the **`display`** property. We will discuss about display property in the next section.

The term “box model” is used when talking about design and layout. Every element that is rendered by the browser is essentially a box. It consists of margins, borders, padding, and the actual content. See the below given picture.



Different Parts Of The Element:

Content area

Content area is at the center of box. It holds the content of the box such as text and images. In the above example text ‘the’ is the content.

Padding

Padding is transparent space surrounding content area. Padding space around the content can be specified by using padding property.

Border

A border that goes around the padding and content can be applied by using border property. Every border is displayed with its specified width, so the

space occupied by the border around padding area depends on the value declared in through the property.

Margin

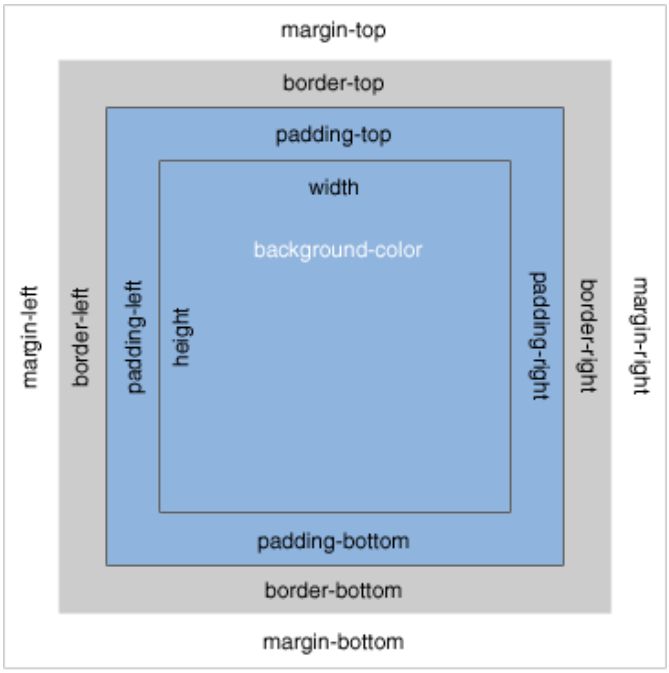
Margin is the transparent area outside the border. Default margin area can be changed using margin property.

The box model allows us to add a border around elements, and to define space between elements. In order to set the width and height of an element correctly in all browsers, you need to know the box model works.

The following sample code explains the idea of different boxes:

```
div {width:300px; padding:25px; border:25px solid navy;
margin:25px 10px 25px 25px;}
```

Refer the below given picture that stages basic style properties associated with box model.



Note important: *When you set the width and height properties of an element with CSS, you just set the width and height of the content area.*

Height And Width Of The Content Area

As mentioned earlier; a block level element always takes up the full width available (stretches out to the left and right as far as it can). Setting the width of a block-level element will prevent it from stretching out to the edges of its container. Then, you can set the margins to auto, to horizontally center the element within its container. The element will take up the special width, and the remaining space will be split equally between the two margins.

HEIGHT

The height property sets the height of the content area. Note that, it doesn't change height of the content. See the below given code:

Sample code:

```
h1{height:5%;background-color:red;text-align:center;}
```

Min-height

The min-height property allows authors to constrain box heights to a certain range.

Max-height

The min-height property allows authors to constrain box heights to a certain range.

WIDTH

The width property sets the width of the content area. Note that, it doesn't change width of the content. Add width property to above code as below:

Sample code:

```
h1{width:10%;height:5%;background-color:red;text-
align:center;}
```

Min-width

The min-width property allows authors of page to constrain content widths

to a certain range.

Max-width

The max-width property allows authors of page to constrain content widths to a certain range.

The properties height, min-height, max-height, width, min-width and max-width set the height/width of the area inside padding, border, and margin of the element. It don't include padding, borders, or margins.

Sample code

```
div {width:500px;height:100px; border:3px;}
```

PADDING

Padding-top

Padding-top property allows the author of page to specify top padding space.

Padding-right

Padding-right property allows the author of page to specify right padding space.

Padding-bottom

Padding-bottom property allows the author of page to specify bottom padding space.

Padding-left

Padding-left property allows the author of page to specify left padding space.
Sample code

```
h1{padding-top:14px;padding-bottom:10px;background-
color:red;}
```

Padding

Padding is the shortcut property used for set all padding properties in one declaration.

The order of values for padding is:

```
padding-top,padding-right,padding-bottom,padding-left
```

Sample code

```
p {padding:2cm 4cm 2cm 4cm;}
```

If two or three values are given, missing values are taken from the opposite side

BORDERS

Border-style

The **border-style** property defines the style of the border. Various styles with its value and the function are given below:

- **None** - no border is drawn
- **Dotted** - the border is a dotted line drawn on top, bottom, left or right of the background of the element
- **Dashed** - the border is a dashed line
- **Solid** - the border is a solid line
- **Double** - the border is a double line
- **Groove** - a 3D groove is drawn in colors based on the 'color' value
- **Ridge** - a 3D ridge is drawn in colors based on the 'color' value
- **Inset** - a 3D inset is drawn in colors based on the 'color' value

Sample code

```
H1 {border-style:solid;}
```

Border-width

The border-width property defines the thickness of the four borders around padding area. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: **thick, medium, or thin.**

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

Sample code

```
H1 {border-width:thick;}

P {border-width:2px 10px 4px 20px;}
```

Border-color

The border-color property is used to set the color of the four borders. The color can be set by:

name – specify a color name, like "red"

RGB – specify a RGB value, like "rgb(255,0,0)"

Hex – specify a hex value, like "#ff3399"

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

If border-color is not set, it inherits the color of the element.

Sample code

```
H1 {border-color:#668fc0;}
```

Border – individual sides

From the examples above you have seen that it is possible to specify a different border for each side. In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left).

Sample code

```
H1 {

    border-top-style:dotted;
    border-bottom-style:solid;
    border-bottom-width:10px;
    border-bottom-color:10px;
   }
```

Border

As you can see from the examples above, there are many properties to consider when dealing with borders. To shorten the code, it is also possible to specify all the individual border properties in one property. The border property is a shorthand property for the following individual border properties:

Border-width

Border-style (required)

Border-color

Sample code

```
H1 {border:5px solid red;}
```

All Border Properties

Border
Border-top
Border-top-width
Border-top-style
Border-top-color
Border-right
Border-right-width
Border-right-style
Border-right-color
Border-bottom
Border-bottom-width
Border-bottom-style

```
Border-bottom-color
Border-left
Border-left-width
Border-left-style
Border-left-color
```

MARGINS

CSS has properties for specifying the margin for each side of an element:

Margin-top

To set top margin for the element

Margin-right

To set right margin for the element

Margin-bottom

To set bottom margin for the element

Margin-left

To set left margin for the element

Sample code

```
h1 {margin-top:2cm;}
p {margin-bottom:50%;}
```

Margin

Margin is the shortcut property to set all margins in one declaration. The order of values for **margin** is:

Margin-top, margin-right, margin-bottom, margin-left

Sample code

```
p {margin:2cm 4cm 2cm 4cm;}

h1 {margin:auto;}
```

All the margin properties can have the following values:

Auto	- the browser calculates the margin
Length	- specify a margin in px,pt, cm, etc.
%	- specifies a margin in % of the width of the containing element
Inherit	- specifies that the margin should be inherited from the parent element

OUTLINE

Outline is like a border surrounded to the border of the element. CSS has different outline properties as it is there to the border.

All outline properties

```
Outline
Outline-top
Outline-top-width
Outline-top-style
Outline-top-color
Outline-right
Outline-right-width
Outline-right-style
Outline-right-color
Outline-bottom
Outline-bottom-width
Outline-bottom-style
Outline-bottom-color
Outline-left
Outline-left-width
Outline-left-style
Outline-left-color
```

Sample code

```
p {outline:1px solid black; border:2px dotted blue;}
```

PART - VII

LAYOUT PROPERTIES

1. Display/Visuality
2. Position
3. Float/Clear

DISPLAY

Display/Visibility

The display property specifies if/how an element is displayed. Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Hiding An Element

Display: none;

Display:none is commonly used with JavaScript to hide and show elements without deleting or recreating them. The <script> element use display:none; as its default. Display:none hides an element, and it will not take up any space. The element will be hidden, and the page will be displayed as if the element is not there:

Visibility:hidden;

Also, visibility property can be set to hidden to hide an element; but it will take up the same space as before. The element will be hidden, but still affect the layout.

Display – Block and Inline

A block element is an element that takes up the full width available, and has a line break before and after it.

An inline element only takes up as much width as necessary, and does not force line breaks.

Override The Default Display Value

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow web standards.

The following example displays <li> elements as inline elements and elements as block elements:

Sample code

```
li {display:inline;}

span {display:block;}
```

Predefined values for Display property

```
Inline,
block,
list-item,
marker,
none.
```

POSITIONING

Position

The positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.

There are four different positioning methods.

Static Positioning

HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.

Static positioned elements are not affected by the top, bottom, left, and right properties.

Fixed Positioning

An element with a fixed position is positioned relative to the browser window, and will not move even if the window is scrolled.

Sample code

```
p.pos_fixed {position:fixed; top30px; right:5px;}
```

Note: IE7 and IE8 support the fixed value only if a !DOCTYPE is specified.

Fixed positioned elements are removed from the normal flow.

Fixed positioned elements can overlap other elements.

Relative Positioning

A relative element is positioned relative to its normal position.

Sample code

```
h2.normalRelative {position:relative;}
h2.left-20Relative {position:relative; left:-20px;}
```

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow. Relatively positioned elements are often used as container blocks for absolutely positioned elements.

Absolute Positioning

An absolute position element is positioned relative to the first element that has a position other than static. If no such element is found, the containing block is <html>.

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.

Absolutely positioned elements can overlap other elements.

Overlapping elements

When elements are positioned outside the normal flow, they can overlap other elements.

The z-index property specifies the stack order of an element (which element should be placed in from of, or behind, the others).

An element can have a positive or negative stack order.

Sample code

```
img {position:absolute; left:0px; top:0px; x-index:-1;}
```

Note: If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.

All CSS Positioning Properties

Bottom – sets the bottom margin edge for a positioned box
Values - auto, length, %, inherit

Clip – clips an absolutely positioned element shape
Values - auto, inherit

Cursor – specifies the type of cursor to be displayed
Values - <url>, auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help

Left – sets the left margin edge for a positioned box
Values - auto, length, %, inherit

Overflow – specifies what happens if content overflows an element's box
Values - auto, hidden, scroll, visible, inherit

Position – specifies the type of positioning for an element
Values - absolute, fixed, relative, static,

inherit

Right - sets the right margin edge for a positioned box

Values - auto, length, %, inherit

Top - sets the top margin edge for a positioned box

Values - auto, length, %, inherit

z-index - sets the stack order of an element

Values - <number>, auto, inherit, none

FLOAT

What is CSS Float?

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

Float is often used with images, but it is also useful when working with layouts.

How elements Float?

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element. The elements after the floating element will flow around it.

The elements before the floating element will not be affected. If an image is floated to the right, a following text flows around it, to the left.

Sample code:

```
Img {float:right;}
```

FLOATING ELEMENTS NEXT TO EACH OTHER

Float

If you place several floating elements after each other, they will float next to each other if there is room.

Here I have made an image gallery using the float property:

Sample code:

```
.gallery{float:left;width:110px;height:90px;margin:5px;}
```

TURNING OFF FLOAT USING CLEAR

Clear

Elements after the floating element will flow around it. To avoid this, use the clear property.

The clear property specifies which sides of an element other floating elements are not allowed.

Add a text line into the above image gallery, using the clear property.

Example:

```
.text_line{clear:both;}
```

All CSS float properties

`Clear` - specifies which sided of an element where other floating elements are not allowed.
Values - left, right, both, inherit, none

`Float` - specifies whether or not a box should float
Values - left, right, both, inherit, none

Reference:

- ***https://developer.mozilla.org***

- ***www.tutorialspoint.com/css/css_tutorial.pdf***
- ***www.w3c.org***
- ***www.quecorp.com***
- ***Special edition using HTML – Sixth edition – by Molly E. Holzsehlog***

DINESH S. ZALMI

HTML

WHAT IS HTML?

HTML is widely used language on the World Wide Web to create web pages. It is based on Standard Generalized Markup Language (SGML). To create a static website HTML is enough. However styles, scripts etc. are used with HTML to make website more smart and presentable.

Hyper Text Markup Language.

Hyper Text: Hyper Text means the text that contains a link within it to jump to another section of the page or to a new page when clicked.

Markup language: A markup language is a programming language that is used to make text more interactive and dynamic. Images, tables, lists, links etc. are displayed using the text-based instructions.

BASIC STRUCTURE OF HTML

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>

    </body>
</html>
```

Elements in Basic Structure

!DOCTYPE :

Since web browsers are executed with special-purpose HTML parsers, rather than general-purpose DTD-based parsers; they don't use DTDs and will never access them even if a URL is provided. DOCTYPE associates a particular SGML or XML document with a Document Type Definition (DTD). DOCTYPE instruction triggers "standards mode" in common browsers.

html :

HTML is composed of a tree of HTML tags, such as text tags. Each tag can have HTML attributes specified. Tags can also have content, including other tags and text. Many HTML tags represent semantics, or meaning. For example, the title tag represents the title of the document.

head :

HTML metadata (data about data) is data about the HTML document. The <head> element is a container for metadata and is placed between the <html> tag and the

<body> tag. Metadata typically define the document title, character set, styles, links, scripts, and other meta information. Metadata is not displayed.

Each <head> element should contain a <title> element indicating the title of the document, although it may also contain any combination of the following elements, in any order:

- Meta : includes information about the document such as keywords and a description, which are particularly helpful for search applications.
- Base : used to create a "base" url for all links on the page.
- Script : used to include JavaScript or VBScript inside the document.
- Object : is designed to include images, JavaScript objects, Flash animations, MP3 files, QuickTime movies and other components of a page.
- Style : used to include CSS rules inside the document.
- Link : used to link to an external file, such as a style sheet or JavaScript file.

title :

Most HTML documents must have a <title> element. However, the <title> element can be omitted in cases where a higher-level protocol provides title information. The title is usually displayed at the top of the browser's title bar.

body :

The HTML <body> tag defines the section of the HTML document or the main content of the HTML document that will be directly visible on your web page.

HTML TAGS

HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties. HTML tag contains three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

WHAT IS SEMANTIC HTML?

Semantic HTML is the use of HTML markup tags to describe the meaning of each element in the document through proper choice of markup tags. Use of semantic HTML makes content readable to both, human and computer.

HTML also contains non-semantic tags like div, span, b, i, u etc. which are non-semantic, meaning these tags cannot be used to describe the meaning of any contents enclosed inside these markup tags; so these tags are to be completely separated to maintain semantic clarity.

Semantic clarity is said to be obtained when semantic tags like h1, p, header, footer, nav, section, article, ul, etc. are well-organized to describe the contents of the webpage.

WHY IS SEMANTIC MARKUP IMPORTANT?

- Semantic HTML is most useful to the **browsers** to give better display on different devices.
- Semantic HTML is a better method for **developers** to distinguish different types of data; and to better able to tie different sources of information together into new web services.
- As the web can be read equally well by both humans and computer, it becomes more accessible since computers are better able to analyse its contents, index it and deliver it. Contents of the document become more meaningful to the visitors like Search engine web crawlers, browser translation tools, or **assistive technologies** such as Speech browsers and screen readers.
- It helps the **web users** like visually impaired people, navigation web users to ascertain proper meaning of the contents.
- **Webpage Ranking to top on search results** get improved thereby bringing the web users closer to your web page. It determines your existence so as to remain competitive.

SEMANTIC ELEMENTS

Various elements displayed in the webpage are individually described with specific meaning like paragraph text, headings to the paragraphs, lists, table contents etc. These elements are denoted by markup tags; and accordingly they are displayed in the browsers by using the default presentation style.

We can group the most common and important semantic markup tags into four categories:

- Document structure tags
- Textual meaning tags
- Multimedia type tags
- Correlation tags

DOCUMENT STRUCTURE TAGS

Div

In the past, the div element was the main way sections of a website were identified and grouped. Div tag is a container that was mostly used with 'id' and 'class' attributes. However, with the release of HTML5, we have several new tags to work with that provide semantic meaning in addition to the grouping attributes offered by the div tag.

HEADER
Site Logo
Heading Elements
Site Navigation
FOOTER
Authorship
Contact
Copyright information
Navigation Links
Back to top
MAIN
Contents unique to page
ASIDE
Page related contents
Advertisements
NAV
Navigation block in:
Header
Footer
Aside
SECTION
Section of MAIN

Header

Header is a container to be used for a web page header which typically contains the site logo, heading elements, and site navigation.

Footer

Footer is a container to be used for a web page footer which typically contains authorship, contact, and copyright information in addition to navigational links and a link back to the top of the web page.

Main

Main is a high-level container to be used to contain all of the content that is unique to a single web page and not repeated across multiple web pages.

Aside

Use to identify content that is related to the main content on the page but not part of the primary flow of the document. For example, the aside element may contain

a glossary definition of a term that appears in a blog post or it may contain advertisements related to the contents of the page.

Nav

Nav is the container to contain blocks of site navigation links. This element is typically placed in the page header and footer, and may also be used in an aside (sidebar) element as well.

Section

The section element is used to mark off sections of a document, such as chapters or major sections of a long form posts.

Article

The <article> element is a good choice to contain entire blog posts, news articles, and similar content. It identify a block of content suitable for reuse and syndication. It should be possible to distribute an article container independently from the rest of the site.

OTHER TAGS

<hr>

TEXTUAL MEANING TAGS

Headings

<h1> is the highest level, or most important heading in a document or in its sections. It is followed by heading levels <h2> through <h6> in order of descending importance.

Paragraphs

The HTML <p> element represents paragraph of text. Headings are often followed by supporting paragraphs.

Strong

Text in any element of the page deserving prominent attention can be marked as strong element. It assigns importance to the contained word or phrase. It also makes the text bold. Strong text helps search engines to index the contents of the page.

Emphasis (em)

Em signifies emphasized contents. It is displayed in italic style. A software reading the text would pronounce the words with a stressed emphasis.

Other Tags

<abbr>, <acronym>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and
.

Example of Semantic Markup of element

```
<!DOCTYPE html>
<html>
    <head>
        <title>Paryavaran'Activities</title>
    </head>
<body>
<article>
<h1>TREE PLANTATION PROGRAMME</h1>
<p>Report of <strong>tree plantation programme</strong> </p>
    <h2>Programme Details</h2>
    <p>Programme was organised at <em>PES Complex</em> on
       29/07/2017 from 9.30 am. </p>
        <h3>Inauguration</h3>
            <p> Programme was inaugurated in the hands of
                Ravi Naik, President of Society. </p>
            <p> V. R. Raik, Principal of HSS, and
                D. S. Jalmi, NSS Programme Officer was present
                for the inauguration programme.</p>
        <h3>Brief report of Plantation Activity</h3>
        <h3>Closing Ceremony</h3>
    <h2>Participants Detail</h2>
</article>
</body>
</html>
```

ts:'

MULTIMEDIA TYPE TAGS

Multimedia can be almost anything you can hear or see. Examples are images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats. Audio, video, and animation have been handled differently by the major browsers. Different formats have been supported, and some formats require extra helper programs (plug-ins) to work.

Audio

Used to add one or more sources of audio content to a document and to allow the browser to pick the best option based on the visitor's device and browser.

The HTML5 <audio> element specifies a standard way to embed a audio in a web page. It supports audios of mp3, wav and ogg audio format. Other formats may require plugins to play audio.

Video

Used to add one or more sources of video content to a document and to allow the browser to pick the best option based on the visitor's device and browser.

The HTML5 <video> element specifies a standard way to embed a video in a web page. It supports videos of mp4, webM and ogg videos format. Other formats may require plugins to play video.

Picture

Image (img)

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. <img> tag is used to insert image in the document. The <img> tag is an empty tag, and require 'src' and 'alt' attribute. HTML5 supports formats like png, jpeg, gif etc.

Following is a simple code used to insert image:

<picture> element

The picture element is used to allow a web browser to pick the best image from the available source options based on the results of a **media** query. It works if at-least one <img> element is placed inside the <picture> container. <img> element should be placed as last element as <source> elements won't work if placed after <img>.

OTHER TAGS

<area>, <map>, <param> and <object>

Examples using multimedia elements:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Paryavaran Activities</title>
    </head>
<body>
        <audio controls="controls" autoplay="autoplay">
                <source src="horse.ogg" type="audio/ogg" />
                <source src="horse.mp3" type="audio/mpeg" />

                Your browser does not support the audio element.
        </audio>

        <video width="320" height="240" controls="controls">

                <source src="movie.mp4" type="video/mp4" />

                <source src="movie.ogg" type="video/ogg" />

                Your browser does not support the video tag.

        </video>

        <picture>

        <picture>

        <source media="(min-width:1500px)" srcset="images/a.png" />

        <source media="(min-width:1200px)" srcset="images/b.png" />

        <img src="images/c.png" alt="a cute kitten" />

        </picture>

</body>
</html>
```

CORRELATION TAGS

Several HTML elements are used to signal a correlation between multiple elements. For example, the use of an ordered list (ol) tells the browser that the items on the list are related to each other and need to appear in a specific order.

Lists

HTML uses following two kinds of lists:

1. Ordered Lists
2. Unordered lists

Ordered Lists

Ordered list element display list of items that start with a number or letter and continue in sequence.

Following options are available to start with sequence:

- Arabic numerals (1,2,3...)
- Upper-case roman numerals (I,II,III...)
- Lower-case roman numerals (i,ii,iii...)
- Upper-case letters (A,B,C...)
- Lower-case letters (a,b,c...)

To define the type of ordered list required, 'type' attribute is placed in <ol> opening tag. Order sequence can be started at by using the 'start' attribute.

Unordered Lists

Unordered list element display the sequence of list items that start with a bullet like a dot, a circle or a square. Following options are available to display sequence:

- Disc (•)
- Circle (o)
- Square (▪)

To define the type of list required, 'type' attribute is placed in <ul> opening tag.

Table

Table is used to arrange data like text, images, links etc. into columns and rows of cells.

- Data is placed in the cells created using opening and closing tag of data cell <td>.
- Data cells are placed in opening and closing tag of <tr> element to form a row of cells.
- Rows are placed in opening and closing tag of <table> element to form a table of rows and columns.
- Data cells with <td> tag can be replaced with <th> tag in the first row if the table contains column headers.
- By default table borders are set to 0, i.e. borders are not shown to the table. To set table border 'border' attribute is used.

EXAMPLE

```
<body>
   <table border="1">
      <tr> <th>State</th>          <th>Language</th> </tr>
       <tr> <td>Goa</td>           <td>Konkani</td>   </tr>
       <tr> <td>Karnataka</td>     <td>Kannada</td>  </tr>
       <tr> <td>Maharashtra</td> <td>Marathi</td>    </tr>
   </table>
</body>
```

Figure

The figure element is used to group together a piece of content, such as an image, chart, graph, or text, and a caption marked off by figcaption tags. By nesting the caption and the content between figure tags a relationship between the nested elements is identified.

Address

This attribute is used to associate contact information with the parent element that contains the address element. For example, when added to an article, the address element provides contact information for the article author, and when added to a web page footer the address identifies contact information for the web page owner.

Anchor - <A>

The HTML anchor tag defines a hyperlink that links one page to another page. The "href" attribute is the most important attribute of the HTML a tag.

Appearance of HTML anchor tag

- An **unvisited link** is displayed underlined and blue.
- A **visited link** displayed underlined and purple.
- An **active link** is underlined and red.

OTHER TAGS

- Link tag - <base>,
- List tags - <dl>, <dt>, <dd>,
- Table tags - tbody, thead, tfoot, col, colgroup and caption
- Form tags - form, input, textarea, select, option, optgroup, button, label, fieldset and legend

CATEGORIES OF ELEMENTS

Container tags and Empty tags

Container Tags

Container tags come in pair. Beginning tag of the container tag is called as opening tag and other tag in the pair with forward slash character '/' is called as closing tag. The opening tag instructs the browser to execute code on the text that follows it. The closing tag is used to mark the end of the code execution. Tag , <h1></h1>, are the examples of container tags. See the example below:

```
<p> My first page. </p>
```

Above given tag is used to mark the document text 'My first page.' as paragraph. In this example, the beginning of the paragraph is marked with tag and marking to specify the end of paragraph is denoted using tag.

Empty tag

Empty tags do not come in pair as they don't have anything to envelop. The examples of empty tags are <hr>,
, etc.

Block-level elements and Inline elements

Block-level elements

Block-level element always starts on a new line and takes up the full width available. It stretches out to the left and right as far as it can by taking up the full width available. element is the best example of block-level element.

Examples of block-level elements:

- <h1>
-
-

Inline elements

Elements produced using inline markup tags does not start on a new line unless it find room for itself to display in the line. The anchor (a) element is the most common inline element, since you use them for links. An inline element can wrap some text inside a paragraph without disrupting the flow of that paragraph.

Example of inline elements:

-
- <a>
- <img>

ATTRIBUTES OF ELEMENTS

An attribute is used to define the characteristics of an element and is placed inside the element's opening tag. They are made up of two parts: a name and a value. Look at the below example:

```
<table border=1>
```

Table tag uses 'border' attribute to define its borders. Value of the attribute is followed by '=' sign. Value '1' sets the thickness of border to 1 pixel.

Element specific attributes

Many HTML tags have a unique set of their own attributes. For example, <img> tag uses 'src', 'alt', 'width' etc.; tag uses 'type', 'start' etc.

```
<ol type="1" start=1101>
```

```
<img src="../images/kite.jpg" alt="Flying Kite Image 200px%280px width" />
```

Global Attributes/Core attributes

Global attributes can be used with almost every HTML Tag in existence. The four core attributes that can be used on the majority of HTML elements (although not all) are:

- id
- title
- class
- style

The id attribute:

The id attribute is used to uniquely identify any element within a page (or style sheet). There are two primary reasons that you might want to use an id attribute on an element:

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.
- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

The title attribute:

The title attribute gives a suggested title for the element. It is often displayed as a tooltip or while the element is loading. The behavior of this attribute will depend upon the element that carries it.

The style attribute:

The style attribute allows you to specify CSS rules within the element. For example:

```
<p style="font-family:arial; color:#FF0000;">Some text..</p>
```

You will learn more about the use of the style attribute in next chapter.

The class attribute:

The class attribute is used to associate an element with a style sheet, and specifies the class of element.

The value of the attribute may be a space-separated list of class names. For example:

```
class="className1 className2 className3"
```

You will learn more about the use of the class attribute in next chapter. So for now you can avoid it.

Internationalization Attributes

There are three internationalization attributes, which are available to most (although not all) elements.

- dir
- lang
- xml:lang

The dir attribute:

The dir attribute is used to specify the direction in which the text should flow. It can take one of two values, as you can see in the table that follows:

- ltr - Left to right (the default value)
- rtl - Right to left (for languages such as Hebrew or Arabic that are read right to left)

For Example:

```
<html dir=rtl>
```

When dir attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

The lang Attribute:

The lang attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the xml:lang attribute in new XHTML documents.

When included within the <html> tag, the lang attribute specifies the language you've generally used within the document. When used within other tags, the lang attribute specifies the language you used within that tag's content. Ideally, the browser will use lang to better render the text for the user.

The xml:lang Attribute:

The xml:lang attribute is the XHTML replacement for the lang attribute. The value of the xml:lang attribute should be an ISO-639 country code. e.g. in, us, uk, etc.

REFERENCE:

- www.googleweblight.com
- www.w3schools.com
- www.yourhtmlsource.com